CLAIMS

What is claimed is:

1. A method comprising:

   (a) providing a connection between a first process and a second process;

   (b) dynamically matching the first process and the second process;

   (c) using a library to dynamically match the first process and the second process; and

   (d) utilizing a flow control provided by a unit.

2. The method of claim 1, wherein the first process provides a push of information to or a pull of information from the second process

3. The method of claim 1, wherein the first process is a client process.

4. The method of claim 1, wherein the second process is a server process.

5. The method of claim 1, wherein the library is an Exigen Object Library (EOL).

6. The method of claim 1, wherein the unit is a transport layer.

7. The method of claim 6, wherein the transport layer supports a plurality of point-to-point connections between the first process and the second process.

8. The method of claim 1, wherein the flow control occurs at a flow origin.

9. The method of claim 8, wherein the flow control backs up information at the flow origin.

10. The method of claims 1, 8 or 9, wherein the flow control prevents an overflow of information to a flow recipient.

11. The method of claim 1, wherein a first name server contains a first plurality of directories.

12. The method of claim 11, wherein the first plurality of directories manage a first plurality of objects.

13. The method of claim 11, wherein one of the plurality of directories is a root of a second name server in order to provide compatibility between the first name server and the second name server.

14. The method of claim 13, wherein the second name server contains a second plurality of directories.

15. The method of claim 14, wherein the second plurality of directories manage a second plurality of objects.

16. A method comprising:

(a) providing a connection between a first process and a second process;

(b) dynamically matching the first process and the second process;

(c) using a library to dynamically match the first process and the second process; and

(d) providing a secure connection between the first process and the second process.

17. The method of claim 16, wherein the first process is a client process.

18. The method of claim 16, wherein the second process is a server process.

19. The method of claim 16, wherein the library is an Exigen Object Library (EOL).

20. The method of claim 16, wherein the first process checks its security rights with a unit.

21. The method of claim 20, wherein the unit is a first name server system.

22. The method of claim 21, wherein the second process checks its security rights with the first name server system.

23. The method of claim 22, wherein the first process provides a push of information to or a pull of information from the second process if appropriate security information has been received from the first name server system.

24. The method of claim 21, wherein the second process checks its security rights with a second name server system.

25. The method of claim 24, wherein the first process provides a push of information to or a pull of information from the second process if appropriate security information has been received from the first name server system and the second name server system.

26. A method comprising:

(a) providing a connection between a first process and a second process;

(b) dynamically matching the first process and the second process;

(c) using a library to dynamically match the first process and the second process; and

(d) asynchronously connecting the first process and the second process.

27. The method of claim 26, wherein the first process provides a push of information to or a pull of information from the second process.

28. The method of claim 26, wherein the first process is a client process.

29. The method of claim 26, wherein the second process is a server process.

30. The method of claim 26, wherein the library is an Exigen Object Library (EOL).

31. The method of claim 27, wherein the connection between the first process and the second process is a single thread that provides an exchange of information.

32. A method comprising:

(a) providing a connection between a first process and a second process;

(b) asynchronously connecting the first process and the second process; and

(c) providing a secure connection between the first process and the second process.

33. The method of claim 32, wherein the first process provides a push of information to or a pull of information from the second process.

34. The method of claim 32, wherein the first process is a client process.

35. The method of claim 32, wherein the second process is a server process.

36. The method of claim 33, wherein the connection between the first process and the second process is a single thread that provides an exchange of information.

37. The method of claim 32, wherein the first process checks its security rights with a unit.

38. The method of claim 37, wherein the unit is a first name server system.

39. The method of claim 38, wherein the second process checks its security rights with the first name server system.

40. The method of claim 39, wherein the first process provides a push of information to or a pull of information from the second process if appropriate security information has been received from the first name server system.

41. The method of claim 38, wherein the second process checks its security rights with a second name server system.

42. The method of claim 41, wherein the first process provides a push of information to or a pull of information from the second process if appropriate security information has been received from the first name server system and the second name server system.